micro:bit treasure hunt

Overview

This activity meets **requirements one**, **two**, and **three** of stage two of the **Digital Maker Staged Activity Badge** (‘Create a piece of interactive or animated digital art using software’, ‘Create a simple computer program to help with a Scouting activity, and try it out’, and ‘Make a simple digital creation that uses code to interact with the wider world through inputs (such as buttons or typing on a keyboard) and outputs (such as a computer screen, sound, or lights)’). It shows people how to create a micro:bit project so they can send ‘warmer’ and ‘colder’ animated graphics between two micro:bits as part of a treasure hunt game.

45 to 90 minutes

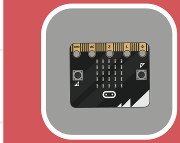
Up to four young people at each computer (pairs are ideal)

If you’re running this activity without internet access, you’ll need to download the software and starter project ahead of time. You’ll probably also need to print handouts.

Wherever you have access to computers. The completed project can be used outdoors.

You will need:

- Laptops or desktop computers (tablets or other mobile devices may work too)
- An activity handout for each group
- micro:bit MakeCode editor
- For each pair:
  - Two micro:bits
  - Two micro:bit battery packs
  - One USB A to micro USB B cable



If your meeting place has internet access

Run the micro:bit MakeCode editor in a web browser at **rpf.io/makecode**.

If your meeting place doesn’t have internet access

Download the micro:bit code editor software before you begin. You can find instructions in the micro:bit guide at **rpf.io/scoutsmicrobit**.

Key messages

- Computers can receive inputs, such as a button being pressed or something being typed on a keyboard, and they can send outputs, such as showing something on a screen, playing sound, or turning on a light. Radio messages are another example of inputs (when received) and outputs (when sent).
- Computers can select instructions to follow, based on inputs they get.
- Computers can repeat instructions.
- MakeCode is a simple drag-and-drop programming language that you can use to make a computer carry out instructions.

micro:bit treasure hunt

Leader instructions

1 Discuss that many digital devices (for example, mobile phones) can use radio transmissions to communicate with each other, and that this is just another method of receiving input or sending output.

2 Discuss how digital devices can improve accessibility. For instance, they can help people who have a hearing impairment or a language barrier.

3 In pairs, the young people should play a quick treasure hunt game:

a. One person should hide an object around the room while the other person isn't looking.
b. The other person has to then find the object.
c. The person who hid the object can only use the words 'warmer' and 'colder' to guide their partner to the object.

4 Explain that everyone will create a digital device that can help people play the game, even when they can't talk to each other.

5 If people haven't used them before, introduce the micro:bits. Explain the micro:bits have two buttons (for input) and a small LED display (for output), and they're also able to communicate with each other using radio waves.

6 Demonstrate how to open the micro:bit MakeCode editor.

7 With the instructions in the printout, the young people will code micro:bits so that they can use them for a treasure hunt.

8 Once their project is complete, the young people can use their paired micro:bits for another treasure hunt, but this time, instead of using the words 'warmer' and 'colder', they'll use their micro:bits to send 'warmer' or 'colder' messages.

Alternatives

- If you don't have access to micro:bits or you've only got a few, young people can still complete the coding section of this activity using the micro:bit simulation in the MakeCode editor. Physical micro:bits are only needed for playing the treasure hunt at the end.

Community and sharing

Young people should have the opportunity to explain their code, or to explain how they found or fixed a bug, to the rest of their team.

Adaptability

- If any young people get stuck, show them the completed code so they can see what their code should look like. You can find the completed project at **rpf.io/treasure-hunt**.
- Consider pairing young people who have no knowledge of the MakeCode editor with those that do.

Safety

If young people are working online, they should ask for permission before viewing any other websites. It's a good idea to set up parental controls — you can find instructions on the NSPCC website (**rpf.io/scouts-nspcc-online**).You should also give each young person a 'Stay safe' leaflet (**rpf.io/scouts-staysafe**).

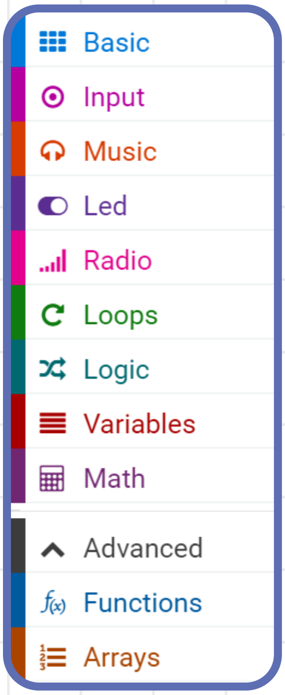
micro:bit treasure hunt

You're going to make a digital device that you can use in a treasure hunt. The device will respond to a signal from another player to show animations telling you whether you're getting 'warmer' or 'colder'. This is great for playing the game when you need to keep quiet, when a player has a hearing impairment, or when you don't speak the same language as the person you're playing with.

Part one: create an image

1 If your device is connected to the internet, open a browser and go to **rpf.io/microbit-new**. If your device isn't connected to the internet, open the MakeCode editor and create a **New Project**.

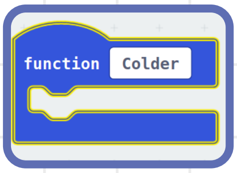
2 Start by making the animation that'll be displayed when the seeker's getting 'colder'. In the menu in the centre, click on **Advanced** and then on `Functions`.



Trivia

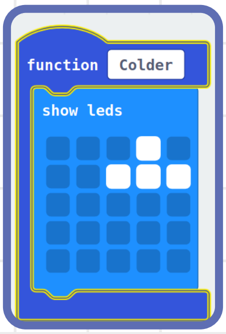
Many mobile devices use Bluetooth radio. Did you know that it's named after the tenth-century Danish king Harald Bluetooth, who united the Danish tribes into a single kingdom? The name was chosen because Bluetooth radio unites communication methods between computers.

3 Create a new function by clicking on **Make a Function**. Call the function `Colder`. A function's just a way of grouping lots of code together.



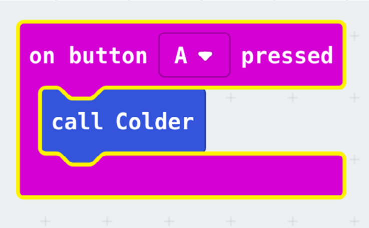
4 You're going to add your animation into this function. Start by clicking the `Basic` button in the menu. Click and drag the `show leds` block into your `Colder` function.

5 You can click on the squares to 'turn them on' and draw a simple design to represent 'colder'. This one's meant to be a cloud, but if you like, you can come up with your own image for 'colder'.

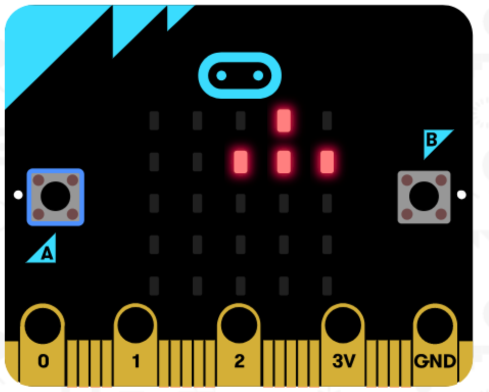


6 To show your image, you can use a button press to call (run) your function. In the `Input` menu, click on the `on button A pressed` block.

7 Go to the `Functions` section, and move the `call function Colder` block so that it is inside the `on button A pressed` block.



8 On the image of the micro:bit that's on the left side of the editor window, click on button A, and your icon should be displayed.

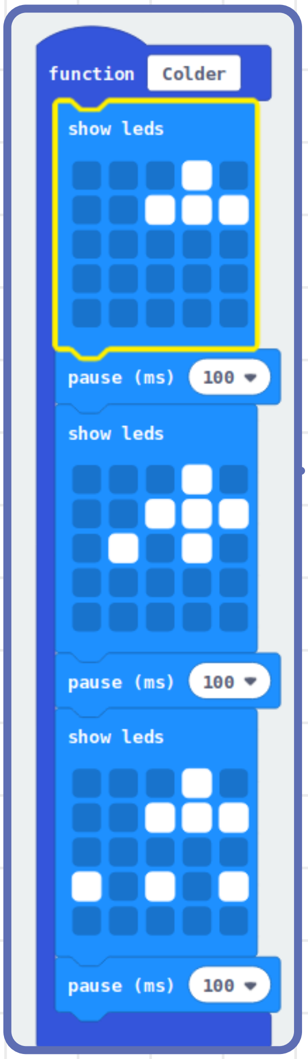


Part two: animate your image

Now you can animate your image so it looks like it's moving. By adding `pause (ms)` blocks between `show leds` blocks, you can create a short animation.

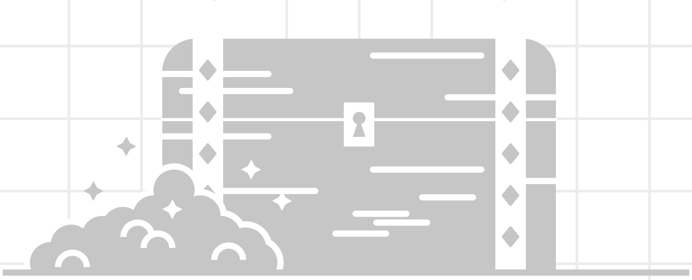
1 Go to the `Basic` section, select the `pause (ms)` block, and insert it below your current `show leds` block.

2 Connect another `show leds` block at the bottom, and create a slightly different image. Then attach another `pause (ms)` block beneath it. You can add as many of these as you want to make an animation that signals 'colder'. Your code will end up looking something like this:

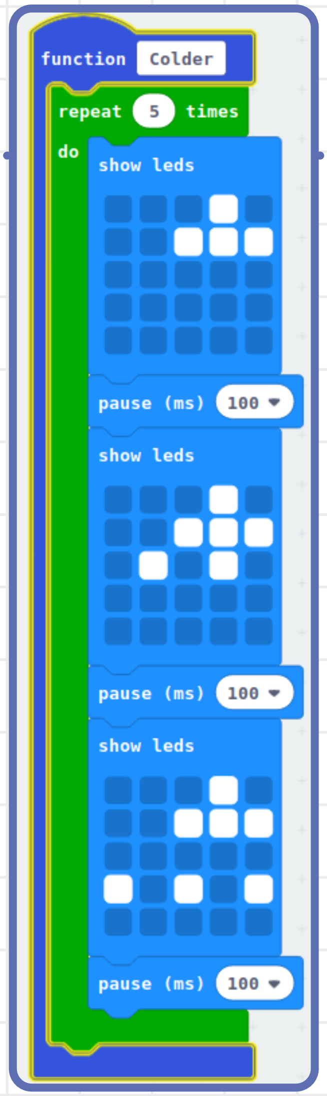


```
function Colder
    show leds
        . . . # .
        . . # # #
        . . . . .
        . . . . .
        . . . . .
    pause (ms) 100
    show leds
        . . . # .
        . . # # #
        . # . # .
        . . . . .
        . . . . .
    pause (ms) 100
    show leds
        . . . # .
        . . # # #
        . . . . .
        # . # . #
        . . . . .
    pause (ms) 100
```

3 Click on **button A** in the image of the micro:bit again to watch your animation.



4 Right now, the animation only plays once – it might look a little better if it repeats. Use a `repeat 5 times` block, found in the `Loops` section, to repeat the animation a few times.



```
function Colder
  repeat 5 times
  do
    show leds
      . . . # .
      . . # # #
      . . . . .
      . . . . .
      . . . . .
    pause (ms) 100
    show leds
      . . . # .
      . . # # #
      . # . # .
      . . . . .
      . . . . .
    pause (ms) 100
    show leds
      . . . # .
      . . # # #
      . . . . .
      # . # . #
      . . . . .
    pause (ms) 100
```

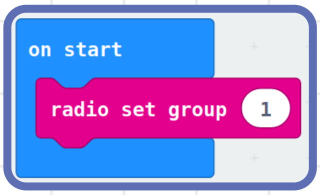
5 Click on **button A** again to view your animation.

6 Now you need a ‘warmer’ animation. This one should be activated by pushing **button B**. Create a new function called `Warmer`, and use it to loop an animation of your own design. To do this, just go through the steps you followed in part one and two to create the `Colder` function and animation.

Part three: make two micro:bits talk to each other

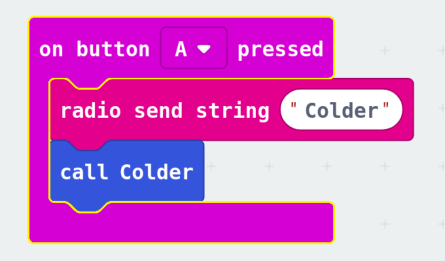
Next you'll use the radio feature of the micro:bit to send instructions from one device to another.

1 To begin, you need to set the **radio number** of the micro:bits. Two micro:bits can only communicate with each other when they're in the same group. Look in the `Radio` section for the block you need:



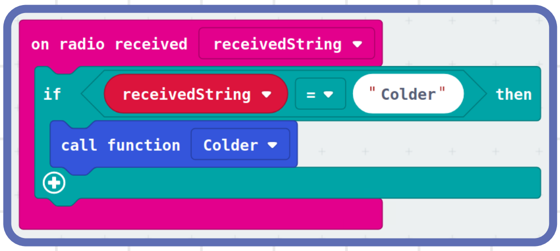
```
on start
  radio set group 1
```

2 When **button A** is pressed, the controller micro:bit should send a message to the receiver micro:bit. The message can just be the word 'Colder'. You'll need blocks from the `Radio` and `Text` sections to create this code.



```
on button A pressed
  radio send string "Colder"
  call Colder
```

3 Whenever the receiver micro:bit gets a message, it needs to check what the text is. If the text is 'Colder', then the micro:bit should call the `Colder` function. Look in the `Logic` and `Radio` sections for the extra blocks you need. You can drag the `Received String` variable out of the `on radio received` block.



```
on radio received receivedString
  if receivedString = "Colder" then
    call function Colder
```

4 Test your code by clicking on **button A** in the image of the micro:bit. Another micro:bit should appear in the window, and both should show the animation.

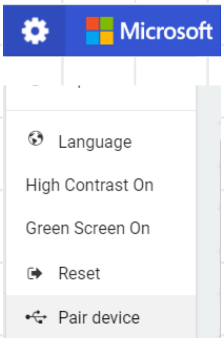
micro:bit treasure hunt

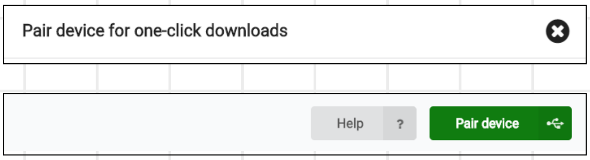
5 Now repeat steps two and three for the ‘Warmer’ animation. When button B is pressed, the controller micro:bit should send the string ‘Warmer’. If the receiver micro:bit gets a message with the string ‘Warmer’, then it should call the `Warmer` function. (Tip: Stack `if then` blocks on top of each other here.)

6 Connect the micro:bit to your computer with a USB cable or via Bluetooth.

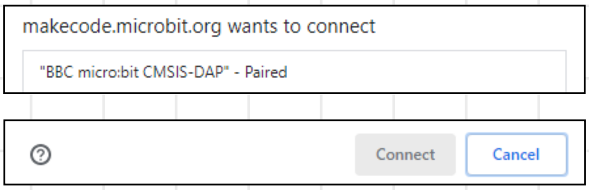
7 Click the ‘Settings’ button and then choose ‘Pair device’.

8 In the next dialogue box, click on ‘Pair device’.



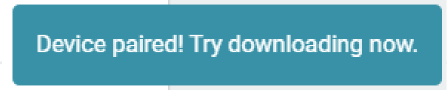

Pair device for one-click downloads

9 Choose the micro:bit that you’re connecting to, then click on ‘Connect’.

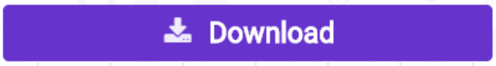

makecode.microbit.org wants to connect

"BBC micro:bit CMSIS-DAP" - Paired

10 If your computer connects successfully, you’ll see the following message:


Device paired! Try downloading now.

11 Now that your micro:bit’s been paired, you can click the ‘Download’ button to transfer your code to your micro:bit.



12 The orange LED on the back of the micro:bit should flash, showing that it’s receiving the program.

13 Repeat this process with the second micro:bit, so both have a copy of the program.

14 Plug both micro:bits into battery packs.

You’re finished – now enjoy your treasure hunt!